

Amendments to the Claims:

This listing of claims will replace all prior versions, and listing, of claims in the application:

Listing of Claims:

1. (currently amended) A computer implemented system employing a central processing unit with access to memory and storage for interpreting application scripts employing a run time program, one or more objects that may or may not have internal properties written to a known specification, and one or more scripts, in which the run time program, the objects, and the scripts can be maintained separately, comprising:

- a) means for dynamically wrapping objects written to a known specification, with additional properties beyond those properties internal to the objects; and
- b) means for utilizing the additional and/or internal properties to modify the execution of the objects.

2. (canceled)

3. (canceled)

4. (previously presented) A computer implemented system employing a central processing unit with access to memory and storage for interpreting application scripts employing a run time program, one or more objects written to a known specification and utilizing one or more scripts, in which the run time program, the objects and the scripts can be maintained separately, comprising:

- a) an interpreting run time program;
- b) scripts that contain property settings for the objects; and
- c) means for utilizing objects by specifying property values according to the script.

5. (previously presented) The system of claim 4 further comprising a means for communicating among objects through the exchange of property values.
6. (previously presented) The system of claim 5 further comprising a means for communicating among objects wherein an event generated by an object triggers an instance of another object.
7. (previously presented) The system of claim 4 further comprising a means for communicating among objects wherein an event generated by an object triggers an instance of another object.
8. (previously presented) A computer implemented system employing a central processing unit with access to memory and storage for interpreting application scripts employing a run time program, one or more objects written to a known specification and utilizing one or more scripts, [it] in which the run time program, the objects and the scripts can be maintained separately, comprising:
- a) an interpreting run time program that has no logical or arithmetic operators;
 - b) scripts that contain property settings for the objects; and
 - c) means for utilizing objects by specifying property values according to the script.
9. (previously presented) The system of claim 8 further comprising a means for communicating among objects through the exchange of property values.
10. (previously presented) The system of claim 9 further comprising a means for communicating among objects wherein an event generated by an object triggers an instance of another object.
11. (previously presented) The system of claim 8 further comprising a means for communicating among objects wherein an event generated by an object triggers an instance of another object.

12. (canceled)

13. (canceled)

14. (canceled)

15. (canceled)

16. (previously presented) The system of claim 1, 4, 5, 6, 7, 8, 9, 10, or 11 further comprising a means for adding additional programming constructs or sub-programs by employing objects that perform the function of programming constructs or sub-programs-wherein expansion of program capabilities is achieved.

17. (canceled)

18. (canceled)

19. (canceled)

20.(currently amended) A computer implemented development and run time system employing a central processing unit with access to memory and storage employing one or more objects that may or may not have internal events written to a known specification which utilizes one or more scripts in which both the objects and the scripts can be maintained separately, utilizing a set of core functionalities comprising:

a) means for dynamically wrapping the objects written to a known specification with additional events beyond those events internal to the objects; [and]

b) means for instantiating objects;

c) means for integrating objects;

d) means for sequencing objects; and

e) means for providing communication among objects

wherein the functionalities performed by the system during execution are determined by the objects used and the scripts.

21. (currently amended) A computer implemented run time system employing a central processing unit with access to memory and storage employing a run time program, one or more objects that may or may not have internal events written to a known specification utilizing a set of core functionalities which interprets one or more scripts, in which the run time program, the objects and the scripts can be maintained separately, comprising:

- a) means for dynamically wrapping the objects written to a known specification with additional events beyond those events internal to the objects; [and]
- b) means for instantiating objects;
- c) means for integrating objects;
- d) means for sequencing objects; and
- e) means for providing communication among objects

wherein the functionalities performed by the system during execution are determined by the objects used and the scripts.

22. (previously presented) A computer implemented system employing a central processing unit with access to memory and storage for employing one or more objects written to a known specification, having property values and event connections, which can be set in time and turned on or off of a visually perceptible display device comprising:

- a) means for setting the values of properties and connecting events;
- b) means for recording and maintaining a history of a plurality of properties settings and event connections as the settings and connections are changed; and
- c) means for traversing the history one change at a time

wherein the property values and event connections may be edited from any point in the history.

23. (canceled)

24. (canceled)

25. (currently amended) A computer implemented system employing a central processing unit with access to memory and storage that interprets one or more scripts, which can be maintained separately, containing property values and event settings that distributes processing to objects that may or may not have internal properties or events written to a known specification, which can be maintained separately, provides and manages data flow among objects, and manages the execution of objects comprising:

- a) means for dynamically wrapping the objects written to a known specification with additional properties and events beyond those properties and events internal to the objects; and
- b) means for utilizing the additional and/or internal properties to manage the execution of the objects and the additional and/or internal events to link and sequence the objects

wherein the run time execution of the objects is determined by property values and events.

26. (currently amended) A computer implemented system employing a central [process] processing unit with access to memory and storage employing two or more objects that may or may not have internal properties or events written to a known specification which implements parallel processing comprising:

- a) means for dynamically wrapping objects written to a known specification with additional properties and events beyond those properties and events internal to the objects;
- b) means for utilizing the additional and/or internal properties to manage the execution of the objects and the additional and/or internal events to link and sequence the objects; and

c) means for specifying the temporal relationship among objects by placing the objects on one or more time lines

wherein execution of the objects occurs concurrently and during which property values may be exchanged among the objects and events may be initiated.

27. (currently amended) An object oriented programming computer implemented system employing a central processing unit with access to memory and storage in which the function of programming constructs is achieved by utilizing objects that may or may not have internal properties or events written to a known specification comprising:

a) means for dynamically wrapping objects written to a known specification with additional properties and events beyond those properties and events provided internal to the object;

b) means for utilizing the additional and/or internal properties and events to link and sequence the objects; and

c) means for specifying a list of property values and event settings

wherein the execution of the objects is determined by the list of property values and event settings.

28. (currently amended) A computer implemented software method for interpreting application scripts employing a run time program, one or more objects that may or may not have internal properties written to a known specification, and one or more scripts, in which the run time program, the objects, and the scripts can be maintained separately, comprising the steps of:

a) dynamically wrapping objects written to a known specification with additional properties beyond those properties internal to the object; and

b) utilizing the additional and/or internal properties to modify the execution of the objects.

29. (canceled)

30. (canceled)

31. (previously presented) A computer implemented software method for interpreting application scripts employing a run time program, one or more objects written to a known specification and utilizing one or more scripts, in which the run time program, the objects and the scripts can be maintained separately, comprising the steps of:

- a) utilizing an interpreting run time program;
- b) utilizing scripts that contain property settings for the objects; and
- c) utilizing objects by specifying property values according to the script.

32. (previously presented) The software method of claim 31 further comprising the step of communicating among objects through the exchange of property values.

33. (previously presented) The software method of claim 32 further comprising the step of communicating among objects wherein an event generated by an object triggers an instance of another object.

34. (previously presented) The software method of claim 31 further comprising the step of communicating among objects wherein an event generated by an object triggers an instance of another object.

35. (previously presented) A computer implemented software method for interpreting application scripts employing a run time program, one or more objects written to a known specification and utilizing one or more scripts, in which the run time program, the objects and the scripts can be maintained separately, comprising the steps of:

- a) utilizing an interpreting run time program that has no logical or arithmetic operators;
- b) utilizing scripts that contain property settings for the objects; and

c) utilizing objects by specifying property values according to the script.

36. (previously presented) The software method of claim 35 further comprising the step of communicating among objects through the exchange of property values.

37. (previously presented) The software method of claim 36 further comprising the step of communicating among objects wherein an event generated by an object triggers an instance of another object.

38. (previously presented) The software method of claim 35 further comprising the step of communicating among objects wherein an event generated by an object triggers an instance of another object.

39. (canceled)

40. (canceled)

41. (canceled)

42. (canceled)

43. (previously presented) The software method of claim 28, 31, 32, 33, 34, 35, 36, 37, or 38 further comprising the step of adding additional programming constructs or sub-programs by employing objects that perform the function of programming constructs or sub-programs wherein expansion of program capabilities is achieved.

44. (canceled)

45. (canceled)

46. (canceled)

47. (previously presented) A computer implemented development and run time software method employing one or more objects for developing and executing an application which utilizes

one or more scripts in which both the objects and the scripts can be maintained separately, and
utilizing a set of core functionalities comprising the steps of:

- a) instantiating objects;
- b) integrating objects;
- c) sequencing objects; and
- d) providing communication among objects

wherein the functionalities performed by the software method during execution are
determined by the objects used and the scripts.

48. (previously presented) A computer implemented run time software method employing one
or more objects written to a known specification for executing an application utilizing a set of core
functionalities which interprets one or more scripts, in which the run time program, the objects and
the script can be maintained separately, comprising the steps of:

- a) instantiating objects;
- b) integrating objects;
- c) sequencing objects; and
- d) providing communication among objects

wherein the functionalities performed by the software method during execution are
determined by the objects used and the scripts.

49. (previously presented) A computer implemented software method for employing one or
more objects written to a known specification, having property values and event connections, which
can be set in time and turned on or off of a visually perceptible display device comprising the steps
of:

- a) setting the values of properties and connecting events;

b) recording and maintaining a history of a plurality of properties settings and event connections as the settings and connections are changed; and

c) traversing the history one change at a time

wherein the property values and event connections may be edited from any point in the history.

50. (currently amended) A computer implemented run time software method employing objects that may or may not have internal properties or events which interprets a script containing property values and event settings, in which the run time program, the objects and the script can be maintained separately, and dynamically executes the objects comprising the steps of:

a) wrapping objects with additional properties and events beyond those properties and events internal to the objects;

b) utilizing the additional and/or internal properties and events to link and sequence the objects; and

c) reading one or more sets of property values and event settings maintained separately from the run time system and the objects

wherein the execution of the objects is determined by the property values and event settings in the script.

51. (previously presented) The software method of claim 50 further comprising the step of adding programming constructs or sub-programs utilizing objects.

52. (currently amended) A computer implemented software method that interprets one or more scripts, which can be maintained separately, containing property values and event settings that distributes processing to objects that may or may not have internal properties or events written

to a known specification, which can be maintained separately, provides and manages data flow among objects, and manages the execution of objects comprising the steps of:

- a) dynamically wrapping the objects written to a known specification with additional properties and events beyond those properties and events internal to the objects; and
- b) utilizing the additional and/or internal properties to manage the execution of the objects and the additional and/or internal events to link and sequence the objects

wherein the execution of the objects is determined by the property values and events.

53. (currently amended) A computer implemented software method employing two or more objects that may or may not have internal properties or events written to a known specification which implements parallel processing comprising the steps of:

- a) dynamically wrapping the objects written to a known specification with additional properties and events beyond those properties and events internal to the object;
- b) utilizing the additional and/or internal properties to manage the execution of the objects and the additional and/or internal events to link and sequence the objects; and
- c) specifying the temporal relationship among objects by placing the objects on one or more time lines

wherein execution of the objects occurs concurrently and during which property values may be exchanged among the objects and events may be initiated.

54. (currently amended) A computer implemented object oriented software programming method in which the function of programming constructs is achieved by utilizing objects that may or may not have internal properties or events written to a known specification comprising the steps of:

- a) dynamically wrapping the objects written to a known specification with additional properties and events beyond those properties and events provided internal to the object;
- b) utilizing the additional and/or internal properties and events to link and sequence the objects; and
- c) specifying a list of property values and event settings
wherein the execution of the objects is determined by the list of property values and event settings.

55. (previously presented) A computer implemented software method for employing one or more standardized objects written to a known specification with properties not internal to the standardized objects comprising the steps of:

- a) using a wrapper object, dynamically wrapping one or more standardized objects with additional properties beyond those properties internal to the one or more standardized objects; and
- b) utilizing the additional and/or internal properties to control the standardized objects.

56. (currently amended) A computer implemented software method for employing one or more standardized objects that may or may not have internal events written to a known specification with events not internal to the standardized objects comprising the steps of:

- a) using a wrapper object, dynamically wrapping one or more standardized objects with additional events beyond those events internal to the one or more standardized objects; and
- b) utilizing the additional and/or internal events to control the one or more standardized objects.

57. (canceled)

58. (currently amended) A computer implemented system employing a central processing unit with access to memory and storage for interpreting application scripts employing a run time program, one or more objects that may or may not have internal events written to a known specification, and one or more scripts, in which the run time program, the objects, and the scripts can be maintained separately, comprising:

- a) means for dynamically wrapping the objects written to a known specification with additional events beyond those events internal to the objects; and
- b) means for utilizing the additional and/or internal events to link and sequence the objects.

59. (previously presented) The system of claim 1 in which the means for dynamically wrapping the objects allows the objects to fully access data outside the system.

60. (currently amended) A computer implemented software method for interpreting application scripts employing a run time program, one or more objects that may or may not have internal events written to a known specification, and one or more scripts in which the run time program, the objects, and the scripts can be maintained separately, comprising the steps of:

- a) using a wrapping object, dynamically wrapping the object written to a known specification with additional events beyond those events internal to the object; and
- b) utilizing the additional and/or internal events to link and sequence the object.

61. (currently amended) A computer implemented run time program method to employ [a] one or more objects that may or may not have internal properties written to a known specification with properties, not native to the objects, specified by one or more scripts comprising the steps of:

- a) using an object available to the run time program, dynamically wrapping the object written to a known specification with additional properties; and
 - b) utilizing the additional and/or internal properties to modify the execution of the object written to a known specification.
62. (previously presented) The method of claim 61 in which the run time program, the objects, and the scripts can be maintained separately.
63. (previously presented) The method of claim 61 in which programming constructs or sub-programs can be implemented in the run time program by utilizing objects that execute the constructs or sub-programs.
64. (previously presented) The method of claim 62 in which programming constructs or sub-programs can be implemented in the run time program by utilizing objects that execute the constructs or sub-programs.
65. (currently amended) A computer implemented run time program method to employ one or more objects that may or may not have internal events written to a known specification with events, not native to the objects, specified by one or more scripts comprising the steps of:
- a) using an object available to the run time program, dynamically wrapping the object written to a known specification with additional events; and
 - b) utilizing the additional and/or internal events to link and sequence the object written to a known specification.
66. (previously presented) The method of claim 65 in which the run time program, the objects, and the scripts can be maintained separately.

67. (previously presented) The method of claim 65 in which programming constructs or sub-programs can be implemented in the run time program by utilizing objects that execute the constructs or sub-programs.

68. (previously presented) The method of claim 66 in which programming constructs or sub-programs can be implemented in the run time program by utilizing objects that execute the constructs or sub-programs.

69. (currently amended) A computer implemented method for interpreting one or more application scripts employing a runtime program comprising the steps of:

a) loading one or more application scripts that can be maintained separately from the runtime program;

b) according to information stored in the application script, loading one or more objects that may or may not have internal properties that can be maintained separately from the runtime program and the scripts;

c) using an object available to the run time program, dynamically wrapping one or more objects written to a known specification with additional properties beyond those properties internal to the object; and

d) utilizing the additional and/or internal properties to modify the execution of the objects based on the values found in the script.

70. (previously presented) The method of claim 69 in which the wrapped objects function as program constructs.

71. (previously presented) The method of claim 69 in which the wrapped objects function as sub-programs.

72. (previously presented) The method of claim 69 in which the wrapped objects access data outside the runtime program.

73. (previously presented) The method of claim 70 in which the wrapped objects access data outside the runtime program.

74. (previously presented) The method of claim 71 in which the wrapped objects access data outside the runtime program.

75. (previously presented) A computer implemented object oriented software programming method in which the function of programming constructs is achieved by utilizing objects written to a known specification comprising the steps of:

a) instantiating a first object that performs the function of a programming construct;

b) utilizing the first object's events to link to one or more additional objects

wherein the order of the execution of the additional objects is determined by the list of property values and event settings in the script and the execution of the first object.

76. (canceled)